

## CLAIMS

1. In a wireless communications device, a method for field diagnosing system software, the method comprising:

5           executing system software;  
          launching a run-time engine; and,  
          processing dynamic instruction sets to field diagnose the system software.

10           2. The method of claim 1 further comprising:  
          in response to field diagnosing the system software,  
          operating on system data and system software; and,  
          following the operating on the system software and system data, executing the system software.

15           3. The method of claim 2 further comprising:  
          forming the system software into symbol libraries, each symbol library comprising symbols having related functionality;  
          arranging the symbol libraries into code sections in a code storage section nonvolatile memory; and,  
20           receiving patch manager run time instructions (PMRTI), including dynamic instruction sets and new code sections, in a file system section nonvolatile memory.

25           4. The method of claim 3 wherein receiving dynamic instruction sets includes receiving diagnosis instruction sets, and wherein receiving a new code section includes receiving a diagnosis code section;

the method further comprising:

storing the diagnosis code sections in nonvolatile memory permanent storage; and,

wherein processing dynamic instruction sets includes

5 processing the diagnosis instruction set to execute the diagnosis code section with the system software.

5. The method of claim 4 wherein processing diagnosis instruction sets includes collecting system data.

10

6. The method of claim 5 wherein processing diagnosis instruction sets includes, in response to executing the diagnosis code section with the system software, collecting system data.

15

7. The method of claim 5 wherein collecting system data includes collecting the addresses and values of symbols in read-write volatile memory.

20

8. The method of claim 5 wherein processing diagnosis instruction sets includes storing the collected system data in a first code section in the file system section.

25

9. The method of claim 5 wherein processing diagnosis instruction sets includes using conditional operation code to analyze the collected data.

10. The method of claim 9 wherein operating on the system data and system software includes updating the system data in response to analyzing the collected data; and,

5 wherein executing the system software includes using the updated system data.

11. The method of claim 6 wherein receiving a diagnosis code section includes receiving predetermined sets of updated system data;

10 wherein processing diagnosis instruction sets includes selecting an updated system data set; and,

wherein operating on the system data and system software includes using the selected updated system data set to execute the system software.

15 12. The method of claim 11 wherein receiving a diagnosis code section includes receiving a test code section having a plurality of temporary code symbol libraries and corresponding constraints;

20 wherein processing diagnosis instruction sets includes executing a first temporary code;

wherein analyzing the collected data includes comparing system data, collected in response to executing the first temporary code, to the corresponding constraints;

25 wherein operating on the system data and system software includes:

temporarily updating the software data per the first temporary code constraints if the collected system data passes analysis; and,

temporarily redirecting selected system software symbols to counterpart symbols in the first temporary code symbol library of the diagnosis code section; and,

wherein processing diagnosis instruction sets includes executing alternate temporary code symbol libraries if the collected system data does not pass analysis.

10

13. The method of claim 12 wherein arranging the symbol libraries into code sections includes starting symbol libraries at the start of code sections and arranging symbols to be offset from their respective code section start addresses;

15

the method further comprising:

storing the start of code sections at corresponding start addresses;

maintaining a code section address table cross-referencing code section identifiers with corresponding start addresses;

20

maintaining a symbol offset address table cross-referencing symbol identifiers with corresponding offset addresses and corresponding code section identifiers; and,

wherein executing temporary code symbol libraries from the test code sections includes updating the symbol offset address table and code section address table with addresses in the diagnosis code section.

25

14. The method of claim 13 wherein receiving a diagnosis code section includes receiving a test code section with temporary code symbol library and constraints organized as system data trigger values; and,

5                    wherein analyzing the collected data includes comparing system data, collected in response to executing the first temporary code, to the sets of system data trigger values.

15. The method of claim 13 further comprising:

10                    transmitting the collection of temporary software data updates and temporarily redirected system software symbols via an airlink interface;

                     receiving an updated code section with an updated code section address table and updated symbol offset address table in the file system section; and,

15                    wherein processing diagnosis instruction sets includes storing the updated code section with updated code section address table and symbol offset address table in permanent storage.

20                    16. The method of claim 8 wherein processing diagnosis instruction sets includes transmitting the collected system data via an airlink interface;

                     the method further comprising:

                     receiving a new patch manager run time instruction with a  
25    new code section including updated data;

replaced a first code section in permanent storage with the  
new code section; and,  
executing the system software using the new code section.

5           17.    The method of claim 3 further comprising:  
              following the field diagnosis of the system software, removing  
the dynamic instructions sets from the file system section.

10           18.    In a wireless communications device, a method for  
field diagnosing system software, the method comprising:  
              executing system software;  
              launching a run-time engine;  
              receiving patch manager run time instructions (PMRTI),  
including dynamic instruction sets and new code sections, in a file system  
15    section nonvolatile memory;  
              processing dynamic instruction sets to field diagnose the  
system software as follows:  
                                  executing the diagnosis instruction sets with the  
system software to collect data;  
20                                analyzing the collected data;  
                                  in response to analyzing the collected data, operating on  
system data and system software; and,  
                                  following the operating on the system software and system  
data, executing the system software.

25

19. In a wireless communications device, a system for field diagnosing system software, the system comprising:

executable system software and system data differentiated into code sections stored in nonvolatile memory permanent storage;

5 dynamic instruction sets for diagnosing the system software in the field; and,

a run-time engine for processing the dynamic instruction sets.

10 20. The system of claim 19 wherein the field diagnosis dynamic instruction sets operate on system data and system software; and,

wherein the system software is executed following the operations on the system software and system data by the dynamic  
15 instruction sets.

21. The system of claim 20 wherein the system software is formed into symbol libraries, each symbol library comprising symbols having related functionality arranged into code sections in a code storage  
20 section nonvolatile memory; and,

the system further comprising:

a file system section of nonvolatile memory receiving patch manager run time instructions (PMRTI), including dynamic instruction sets and new code sections.

25

22. The system of claim 21 wherein the file system section receives a diagnosis instruction set and a diagnosis code section;

wherein the diagnosis code section is stored in nonvolatile memory; and,

5 wherein the diagnosis instruction set executes the diagnosis code section with the system software.

23. The system of claim 22 wherein the diagnosis instruction sets collect system data.

10

24. The system of claim 23 wherein the diagnosis instruction sets collect system data in response to executing the diagnosis code section with the system software.

15

25. The system of claim 23 further comprising:  
read-write volatile memory; and,  
wherein the diagnosis instruction sets collect the addresses and values of symbols stored in read-write volatile memory.

20

26. The system of claim 23 wherein the diagnosis instruction sets store the collected system data in a first code section in the file system section.

27. The system of claim 23 wherein the diagnosis  
25 instruction sets use conditional diagnosis instruction sets to analyze the collected data.



28. The system of claim 27 wherein the system data is updated in response to analyzing the collected data, and the system software is executed using the updated system data.

5

29. The system of claim 24 wherein the diagnosis code section includes predetermined sets of updated system data;

wherein the diagnosis dynamic instruction sets select an updated system data set; and,

10

wherein the system software executes using the selected updated system data set.

15

30. The system of claim 29 wherein the diagnosis code section includes a plurality of temporary code symbol libraries and corresponding constraints;

wherein the diagnosis instruction sets execute a first temporary code and compare system data, collected in response to executing the first temporary code, to the corresponding constraints;

20

wherein the system data is temporarily updated per the first temporary code constraints if the collected system data passes analysis, and wherein the system software temporarily redirects selected system software symbols to counterpart symbols in the first temporary code symbol library of the diagnosis code section; and,

25

wherein the diagnosis instruction sets execute alternate temporary code symbol libraries if the collected system data does not pass analysis.

31. The system of claim 30 wherein the system software includes symbol libraries starting at the start of code sections, symbols arranged to be offset from their respective code section start addresses, and the start of code sections being stored at corresponding start addresses;

the system further comprising:

a code section address table cross-referencing code section identifiers with corresponding start addresses;

10 a symbol offset address table cross-referencing code section identifiers with corresponding offset addresses and offset addresses and corresponding code section identifiers; and,

wherein the diagnosis instruction sets update the symbol offset address table and code section address table with addresses in the diagnosis code section.

32. The system of claim 31 wherein the diagnosis code section includes constraints organized as system data trigger values;

20 wherein the diagnosis instruction sets analyze the collected data by comparing system data, collected in response to executing the first temporary code, to the sets of system data trigger values.

33. The system of claim 31 further comprising:

an airlink interface;

25 wherein the diagnosis instruction sets transmit the collection of temporary software data updates and temporarily redirected system

software symbols, via the airlink interface, and receive an updated code section with an updated code section address table and updated symbol offset address table in the file system section via the airlink interface; and,

5                    wherein the diagnosis instruction sets store the updated code section with updated code section address table and symbol offset table in permanent storage.

34.    The system of claim 26 further comprising:  
10                    an airlink interface to transmit the system data collected by the diagnosis instruction sets; and,  
                      a new patch manager run time instruction with a new code section including updated data received via the airlink interface;  
                      wherein diagnosis instruction sets replace a first code section  
15                    in permanent storage with the new code section; and,  
                      wherein the system software is executed using the new code section.

35.    In a wireless communications device, a system for field  
20                    diagnosing system software, the system comprising:  
                      executing system software and system data differentiated into code sections stored in nonvolatile memory permanent storage;  
                      a file system section of nonvolatile memory receiving patch manager run time instructions (PMRTI), including dynamic instruction  
25                    sets and new code sections;

a run-time engine for processing the dynamic instruction sets; and,

wherein the dynamic instruction sets diagnose the system software in the field by collecting and analyzing data, and operate on  
5 system data and system software in response to analyzing the collected data.

10  
15  
20  
25  
30  
35  
40  
45  
50  
55  
60  
65  
70  
75  
80  
85  
90  
95  
100